

# Objavljivanje Python web aplikacije

Nakon što smo kreirali aplikaciju i očistili je od grešaka, možemo je i postaviti na server na internetu. Pošto se većina rešenja za zauzimanje web servera i domena plaća, iskoristićemo servis koji će nam omogućiti da objavimo svoju aplikaciju na serveru, ali uz odgovarajuće limite. Platforma koja nam će to omogućiti se zove <https://www.pythonanywhere.com/>. Neka od ograničenja za besplatan nalog su:

- mogućnost objavljivanja samo jedne aplikacije sa domenom koji će se voditi na korisničko ime naloga (ako prilikom kreiranja naloga izaberemo korisničko ime student\_2020, link ka našoj aplikaciji će izgledati ovako: [http://student\\_2020.pythonanywhere.com/](http://student_2020.pythonanywhere.com/));
- nemogućnost SSH pristupa nalogu;
- ograničen pristup eksternim sajtovima;
- mogućnost zakazivanja jednog zadatka;
- nemogućnost izvršavanja procesa koji konstantno rade.

Više informacija o dostupnim planovima može se pronaći na [zvaničnoj stranici](#).

U ovoj nastavnoj jedinici pokazaćemo kako da kod iz prethodnog modula, odnosno Django projekat, objavimo upravo na ovom sajtu.

Prvo je potrebno kreirati nalog na sajtu [pythonanywhere.com](https://www.pythonanywhere.com/). Otvoriti stranicu <https://www.pythonanywhere.com/pricing/> i odabrati *Create Beginner Account*:

pythonanywhere

## Plans and pricing

**Beginner: Free!**

A limited account with one web app at your-username.pythonanywhere.com, restricted outbound Internet access from your apps, low CPU/bandwidth, no IPython/Jupyter notebook support.  
It works and it's a great way to get started!

[Create a Beginner account](#)

**Education accounts**

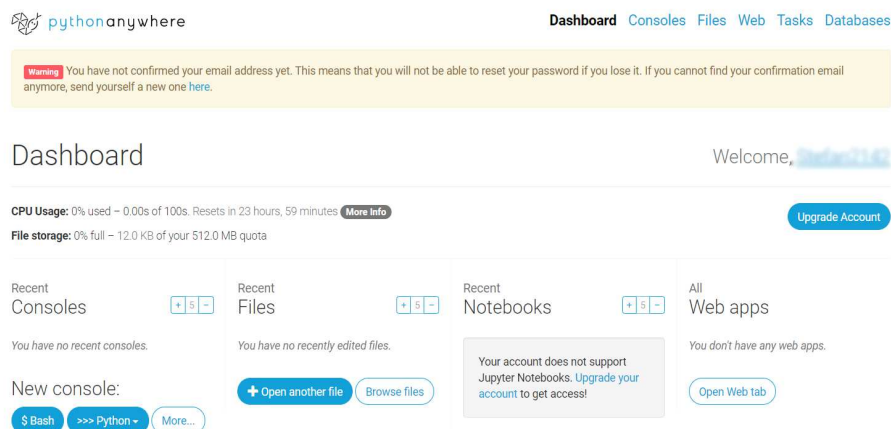
Are you a teacher looking for a place your students can code Python? You're not alone. Click through to find out more about our [Education beta](#).

All of our paid plans come with a no-quibble 30-day money-back guarantee – you're billed monthly and you can cancel at any time. The minimum contract length is just one month. You get unrestricted Internet access from your applications, unlimited in-browser Python, Bash and database consoles, and full SSH access to your account. All accounts (including free ones) have screen-sharing with other PythonAnywhere accounts, and free SSL support (though you'll need to get a certificate for your own domains).

Hacker	\$5/month	Web dev	\$12/month	Startup	\$99/month	Custom	\$5 to \$500/month
Run your Python code in the cloud from one web app and the console		If you want to host small Python-based websites for you or for your clients		Start a business and don't worry about having to scale to handle traffic spikes		Want a combination that's not on the list? Create your own! All custom plans have:	
A Python IDE in your browser with unlimited Python/bash consoles		A Python IDE in your browser with unlimited Python/bash consoles		A Python IDE in your browser with unlimited Python/bash consoles		A Python IDE in your browser with unlimited Python/bash consoles	
One web app on a custom domain or		Up to 2 web apps on custom domains		Up to 3 web apps on custom domains		Up to 20 web apps, on custom	

Slika 16.1. Izgled početne strane [www.pythonanywhere.com](https://www.pythonanywhere.com/)

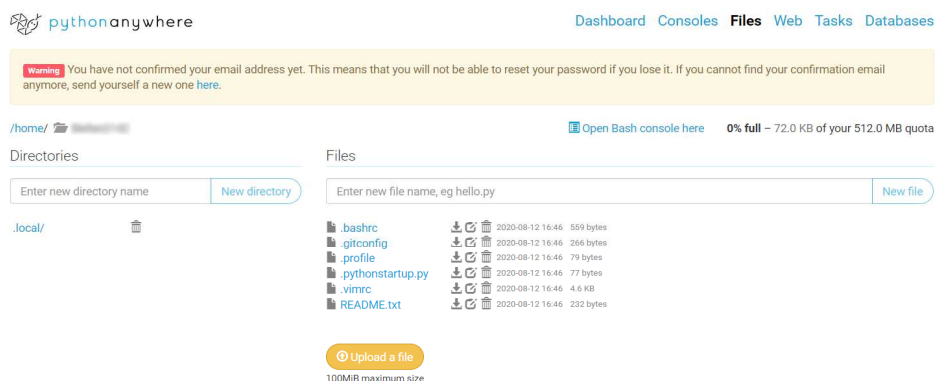
Nakon što smo uspešno kreirali nalog, pokazaće nam se prikaz glavne stranice kao na sledećoj slici.



Slika 16.2. Stranica nakon uspešnog kreiranja naloga

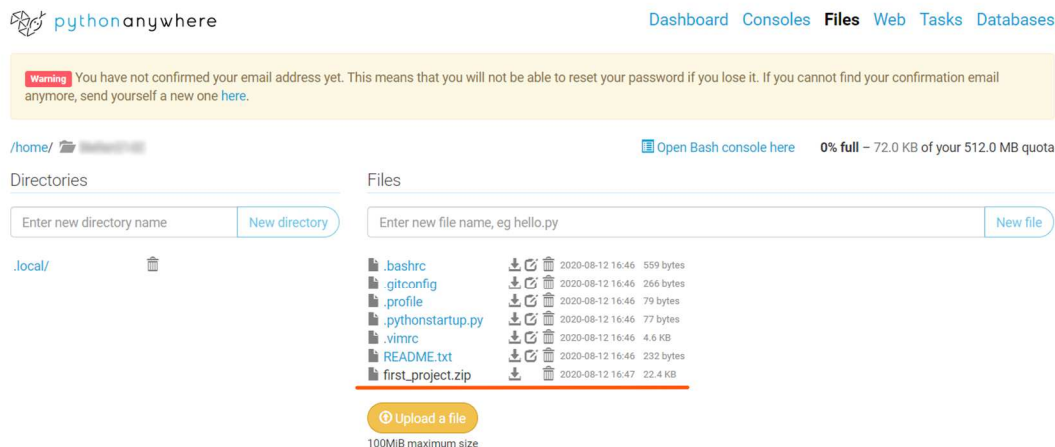
Sad je potrebno zipovati (arhivirati) naš projekat pre nego što ga prebacimo na PythonAnywhere. Ovo ćemo učiniti desnim klikom na glavni folder našeg projekta i izabrati, na Windows platformi, Send To -> Compressed (Zipped) Folder. Fajl koji nastane nakon te kompresije nazvati first\_project.zip. Dakle, treba zipovati folder koji sadrži manage.py fajl zajedno sa first\_project i book\_library i ostalim folderima. Da li smo zipovali pravi folder – možemo proveriti duplim klikom na first\_project.zip; ako u njemu prvo i jedino što vidimo jeste first\_project folder (koji nakon što se otvori sadrži first\_project, book\_library, manage.py...) – onda smo ispravno ispratili instrukcije.

Pošto smo zipovali fajl, on je spreman za prebacivanje na server. Na interfejsu sa slike 16.2. potrebno je kliknuti na dugme *Browse Files*, koje će nas odvesti na Files tab, na stranicu koja izgleda ovako:



Slika 16.3. Prikaz sadržaja našeg servera na Files tabu

Izabrati *Upload a file* i iz otvorenog prozora izabrati naš `first_project.zip` fajl. Nakon uspešnog prebacivanja fajla, direktorijum našeg servera treba izgledati ovako:



Slika 16.4. Prikaz fajlova nakon pravilnog otpremanja `first_project.zip` fajla

Pošto smo dodali naš zipovani projekat, treba ga i otpakovati. PythonAnywhere nam tu funkcionalnost pruža u konzoli i to nakon što kliknemo na *Open Bash console here*. U tom trenutku u pregledaču se otvara novi tab, koji predstavlja konzolu kroz koju možemo upravljati i manipulirati folderima i fajlovima na našem serveru. U toj konzoli treba prvo kreirati virtuelno okruženje komandom:

```
mkvirtualenv django20 --python=/usr/bin/python3.6
```

Na ovaj način kreiramo virtuelno okruženje za rad. Virtuelna okruženja nisu specifična samo za tu konzolu i server, već se mogu koristiti i lokalno na našim mašinama. Ta okruženja zapravo predstavljaju zasebne direktorijume na disku (u ovom slučaju će se taj direktorijum zvati `django20`). Upotrebna vrednost virtuelnih okruženja se bazira na tome da izolujemo radno okruženje od glavnog instalacijskog direktorijuma Pythona, kako bismo u radnom okruženju mogli slobodno da instaliramo neophodne verzije paketa i biblioteka, jer će se često dešavati da radimo na projektima koji zahtevaju baš određene verzije paketa. Pa kako ne bismo konstantno brisali i instalirali neophodne pakete iz glavnog instalacijskog direktorijuma, virtuelna okruženja nam daju poseban, izolovani direktorijum za rad.

## Pitanje

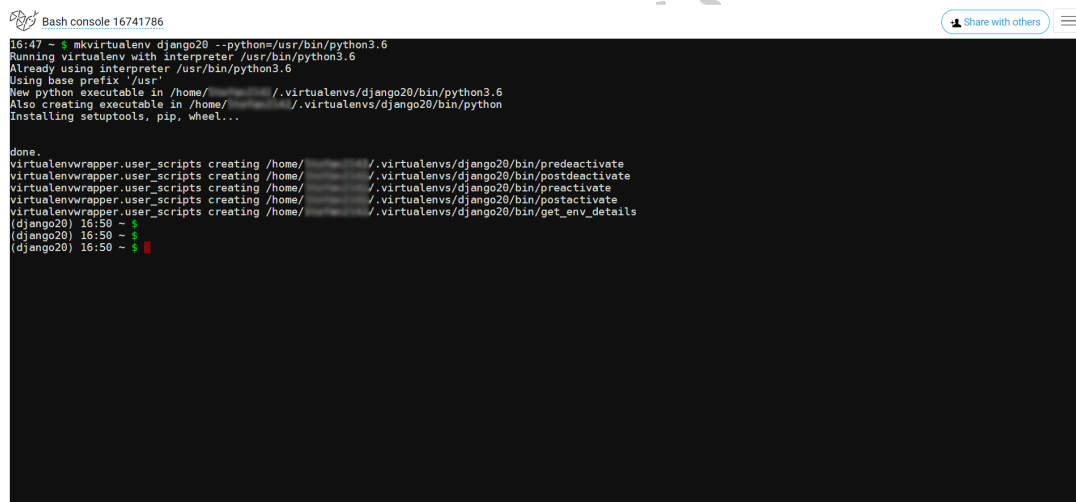
Da li je bitno da komandu `mkvirtualenv` pokrenemo upravo iz Bash konzole PythonAnywhere okruženja ili je možemo pokrenuti i iz našeg komandnog prompta?

- **Moramo je pokrenuti iz Bash konzole**
- Svejedno je

## Objašnjenje:

*Tačan odgovor je da je moramo pokrenuti upravo iz Bash konzole PythonAnywhere okruženja, jer prvo – želimo da kreiramo virtuelni direktorijum za rad na tom serveru (sa komandnom linijom radimo na svojoj lokalnoj mašini), i drugo – ta komanda je svojstvena operativnim sistemima baziranim na Linuxu i ne bi bila prepoznata na Windowsu.*

Na sledećoj slici možemo videti kako izgleda primer uspešnog pokretanja te komande. Obratite takođe pažnju na tekst na početku svake linije koji se nalazi u okviru običnih zagrada – koji i predstavlja ime našeg virtuelnog okruženja:



```
16:47 ~$ mkvirtualenv django20 --python=/usr/bin/python3.6
Running virtualenv with interpreter /usr/bin/python3.6
Already using interpreter /usr/bin/python3.6
Using base prefix '/usr'
New python executable in /home/.../.virtualenvs/django20/bin/python3.6
Also creating executable in /home/.../.virtualenvs/django20/bin/python
Installing setuptools, pip, wheel...
done.
virtualenvwrapper.user_scripts creating /home/.../.virtualenvs/django20/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/.../.virtualenvs/django20/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/.../.virtualenvs/django20/bin/preactivate
virtualenvwrapper.user_scripts creating /home/.../.virtualenvs/django20/bin/postactivate
virtualenvwrapper.user_scripts creating /home/.../.virtualenvs/django20/bin/get_env_details
(django20) 16:50 ~$
```

Slika 16.5. Ispis PythonAnywhere konzole nakon pokretanja `mkvirtualenv`

U ispisu komande `dir` možemo videti i ime našeg zipovanog projekta. Otpakovaćemo ga komandom `unzip first_project.zip`. Pošto server završi sa otpakivanjem fajla, u toj istoj konzoli pokrenut i `pip install django`. Tačnije, u ovom koraku instaliramo sve biblioteke potrebne za naš projekat. Pošto se naša aplikacija zasniva samo na Djangu, samo taj paket nam je i potreban. Nakon završene instalacije Djanga i otpakivanja projekta u folder na serveru, možemo nastaviti sa podešavanjem servera.

Sa početne stranice PythonAnywhere sajta (strana koja sadrži Dashboard i ostale tabove i informacije) izabrati *Web* tab i u njemu kliknuti na *Add a new web app* kao na slici:

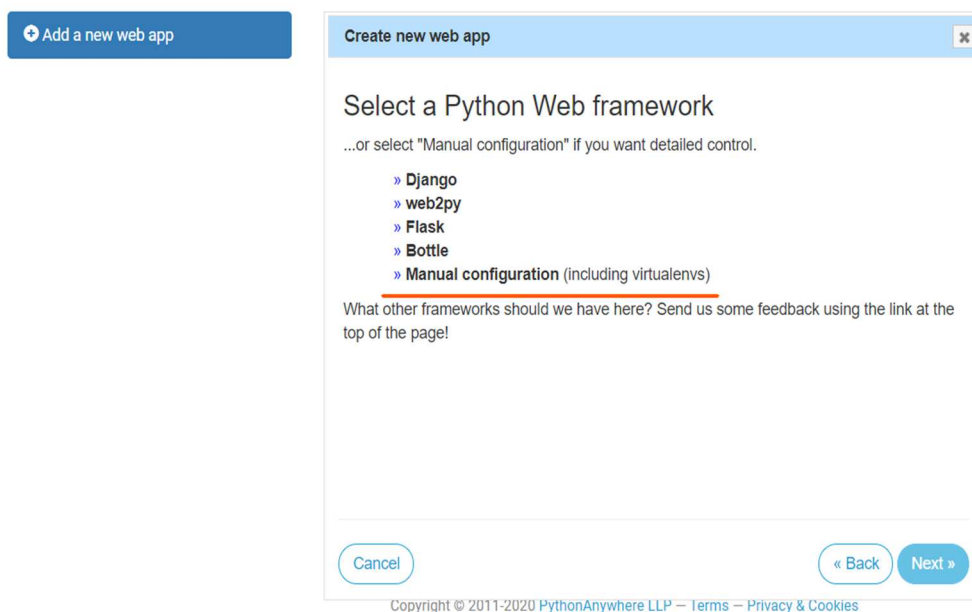
[+ Add a new web app](#)

You have no web apps

To create a PythonAnywhere-hosted web app, click the "Add a new web app" button to the left.

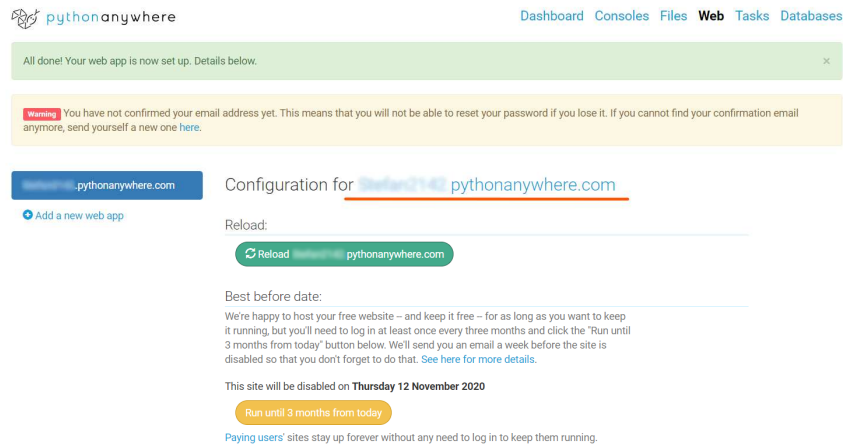
Slika 16.6. Stranica taba Web za kreiranje pythonanyhwere aplikacije

Nakon ispisa informacije da ne možemo izabrati proizvoljno ime i domen aplikacije, kliknuti *Next*. Na sledećoj strani među ponuđenim opcijama izabrati *Manual configuration (including virtualenvs)*:



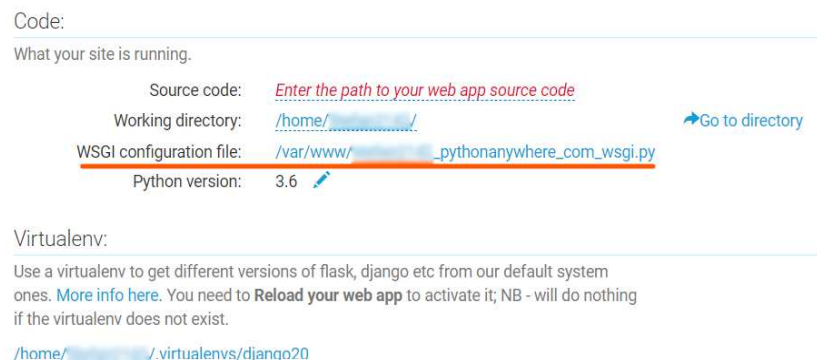
Slika 16.7. Drugi korak pri kreiranju pythonanyhwere aplikacije

Na strani koja se sledeća prikaže izabrati Python 3.6 kao verziju. Taj izbor će nas prebaciti na sledeću stranicu, koja se odnosi na informacije o manuelnoj konfiguraciji servera, gde treba izabrati *Next*. Nakon toga će nam se otvoriti stranica sa našom aplikacijom, gde je možemo dalje podešavati. Link ka našoj aplikaciji je pri vrhu stranice:



Slika 16.8. Uspešno kreirana PythonAnywhere aplikacija

Iako smo postavili projektni folder naše aplikacije na server i kreirali aplikaciju na serveru – nismo ih još uvek povezali pa tako, ako bismo ispratili podvučeni link na prethodnoj slici otvorila bi se generička *Hello, World!* stranica. Za konfiguraciju treba promeniti par stvari u našoj PythonAnywhere aplikaciji. Za početak – na stranici iz prethodne slike naći sekciju o kodu 'Code' i link ka WSGI fajlu:



Slika 16.9. Polje sa putanjom ka serverskom WSGI fajlu

Kliknuti na link koji vodi ka tom fajlu. U novom tabu pregledača će nam se otvoriti sadržaj ovog fajla. U njemu je trenutno kod koji pokreće generičku početnu Hello World! stranicu. Iz tog fajla obrisati sve osim sekcije koja počinje sa `# ++++++++ DJANGO ++++++++`, dok u toj sekciji treba otkomentarizirati samo linije sa kodom. Nakon toga, treba izmeniti promenljivu `path` tako da pokazuje na korenski direktorijum našeg projekta (putanja korenskog direktorijuma na serveru) koji smo prethodno raspakovali, kao i podesiti `['DJANGO_SETTINGS_MODULE']` promenljivu tako da pokazuje u kom folderu naše Django aplikacije se nalazi `settings.py` fajl. Pa tako, na kraju, WSGI fajl treba da izgleda ovako:

Slika 16.10. Izgled WSGI fajla nakon izmena

Nakon što smo završili sa svim potrebnim izmenama – sačuvati fajl klikom na *Save* u gornjem desnom uglu taba i zatvoriti taj tab. Sa stranice naše PythonAnywhere aplikacije pronaći sekciju Virtualenv (odmah nakon sekcije Code) i dodati putanju do virtuelnog okruženja klikom na *Enter path to a virtualenv, if desired*, koji smo prethodno preko konzole kreirali pomoću komande mkvirtualenv.

#### Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[Enter path to a virtualenv, if desired](#)

Slika 16.11. Virtualenv sekcija stranice PythonAnywhere aplikacije

U tekstualnom polju koje se pojavljuje kucamo ime svog virtuelnog okruženja, a to je django20, i to potvrđujemo klikom na dugme pored. Ostalo je još da u podešavanjima Django aplikacije proverimo da li se našoj aplikaciji može pristupiti sa bilo kog domena tako što ćemo otvoriti Files tab naše pythonaynwhere aplikacije, navigirati ka first\_project/first\_project/ direktorijumu i kliknuti na settings.py fajl kao što je to učinjeno na slici:



```

/home/Stefan2142/first_project/first_project/settings.py
1 """
2 Django settings for first_project project.
3
4 Generated by 'django-admin startproject' using Django 3.0.8.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.0/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.0/ref/settings/
11 """
12
13 import os
14
15 # Build paths inside the project like this: os.path.join(BASE_DIR, ...)
16 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
17
18 # Quick-start development settings - unsuitable for production
19 # See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/
20
21 # SECURITY WARNING: keep the secret key used in production secret!
22 SECRET_KEY = '24t11^6+6tgm_sm_njp6*i9(jesr%bv3^#)*jx9*rwq&rml6ie'
23
24 # SECURITY WARNING: don't run with debug turned on in production!
25 DEBUG = True
26
27
28 # ALLOWED_HOSTS = ['*']
29 ALLOWED_HOSTS = ['*']
30
31
32 # Application definition
33

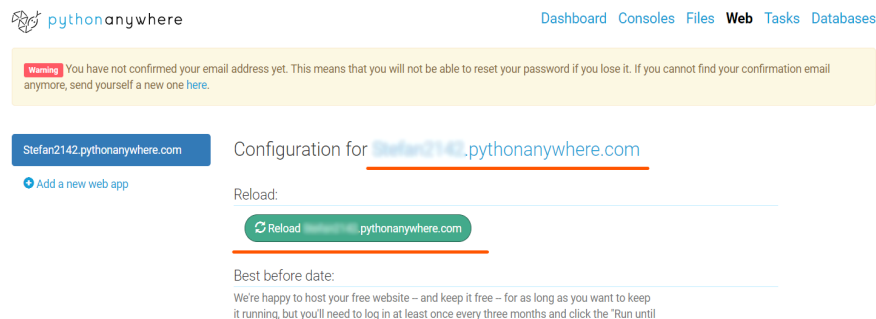
```

Slika 16.12. Deo izgleda settings.py fajla koji se odnosi na ALLOWED\_HOSTS listu

U fajlu koji se otvori, proveriti da li je lista ALLOWED\_HOSTS ovako definisana:

```
ALLOWED_HOSTS = ['*']
```

Ako nije, ispraviti da izgleda tako i sačuvati fajl klikom na *Save* u gornjem desnom uglu. Vratiti se na Web tab naše aplikacije i kliknuti na *Reload* <username>.pythonanywhere.com. (deo <username> se odnosi na korisničko ime koje smo izabrali prilikom kreiranja naloga). Sačekati da se osvežavanje aplikacije završi i otvoriti link ka našoj aplikaciji.



Slika 16.13. Poslednji korak, klik na Reload

Ono što vidimo je početni pogled koji smo definisali u Django aplikaciji, kao i kada smo joj pristupali kroz lokalnu adresu. I ostali linkovi koje smo koristili na lokalnoj IP adresi, kao što su 127.0.0.1:8000/, 127.0.0.1:8000/books, 127.0.0.1:8000/add-book i ostali, rade i na PythonAnywhere platformi. Prednost ovog pristupa je to što našoj aplikaciji možemo pristupiti sa bilo kog uređaja, kao i to što dokle god opet ne kliknemo na osvežavanje aplikacije – Reload, sve knjige koje dodajemo biblioteci ostaju sačuvane i nakon što isključimo pregledač.



Takođe je važno napomenuti da pythonanyhwere nije jedini servis koji nudi ograničeno besplatno objavljivanje aplikacija, već postoje i drugi kao što su Heroku, AlwaysData...

## Rezime

- Između mnoštva platformi i rešenja odlučili smo se za korišćenje <https://www.pythonanywhere.com/> platforme za objavljivanje naše Django aplikacije. Ta platforma nam pruža mogućnost objavljivanja samo jedne aplikacije, sa domenom koji će se voditi na korisničko ime naloga (ako prilikom kreiranja naloga izaberemo korisničko ime student\_2020, link ka našoj aplikaciji će izgledati ovako: [http://student\\_2020.pythonanywhere.com/](http://student_2020.pythonanywhere.com/)) (link nije dostupan jer je reč o primeru).
- Nakon što zipujemo svoj projekat i otpremimo ga na PythonAnywhere server, iz Bash konzole je potrebno kreirati i virtuelno okruženje, komandom `mkvirtualenv django20 --python=/usr/bin/python3.6`.
- Nakon kreiranja virtuelnog okruženja potrebno je izvršiti izmene u settings.py fajlu Django aplikacije i WSGi fajlu pythonanyhwere servera, kako bi se upravo toj aplikaciji na serveru moglo pristupiti sa interneta, preko definisanog linka aplikacije.

